UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/778,087	02/07/2001	Takahiro Ishikawa	1538.1010	4567

21171 7590 12/29/2005

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 12/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/778,087	ISHIKAWA ET AL.	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 October 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is responsive to Applicant's submission filed on October 3, 2005. Claims 1-12 are pending.

Response to Amendment

2. It is noted that the amendment to the claims does not comply with the requirements of 37 CFR 1.121(c) because the status of claim 7 is not "original" as Applicant indicates (amendment, page 4), but is rather --previously presented--.

Response to Arguments

3. Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that Martin fails to disclose, "a certain class-type variable is contained in an execution statement to be executed in parallel or a certain class-type variable is specified in a parallelization directive as a class to be parallelized," as recited in amended claim 1, or in other words, that Martin does not include any class-type variables (remarks, page 6).

However, Martin does in fact disclose class-type variables (see, for example, FIG. 9 and column 7, lines 34-39, noting the C++ source code that "declares a class type"). Martin further discloses that certain class-type variables are specified in parallelization directives (see, for example, column 7, line 50 to column 8, line 5, noting that the parallelization directive includes the "number of objects ... expected in a given class"). The class is to be parallelized and executed in parallel (see, for example, column 9, lines 48-54).

Moreover, Meyer teaches that in the C++ language, "objects are assumed to be variables of a class type and are referred to as instances of the class" (see, for example, column 1, lines 22-25). In other words, like Martin, Meyer too discloses class-type variables.

Applicant contends that Meyer and Martin do not disclose, "detecting that a certain class-type variable is contained in an execution statement to be executed," as recited in amended claim 1, because the calls disclosed in Meyer are calls for messages sent from one process to another process (remarks, page 6).

However, in Meyer, the calls for messages sent from one process to another process are in fact calls concerning "the distribution of the application over [those] processes" (see, for example, column 6, lines 1-10). Regardless, Meyer further discloses that certain class-type variables are to be distributed and executed over several processes, and that the class-type variables to be distributed are in fact detected (see, for example, column 5, lines 28-45, noting that the "class declarations of the sources of the application are analyzed" in one step of a process for "adapting an object-oriented application [that] can be distributed over a plurality of operating system processes").

Moreover, Martin discloses that the certain class-type variables to be executed in parallel are in fact detected (see, for example, column 9, lines 44-47, noting that the pre-compiler is "arranged to read the required availability for a class of objects and ... to read the average performance time indicated for objects of the class").

Applicant contends that Meyer fails to disclose, "generating an instruction to call [a] construction instruction routine for an object of the class ... and generating an instruction to call

a destruction instruction [routine] for the generated object of the class," as recited in claim 1, because in Meyer, the stub object does not have the same function as the real object, and does not correspond to the recited "object of the class" (remarks, page 7).

However, the examiner does not agree with Applicant's characterizations. Meyer discloses that the "stub objects should behave towards the other objects in exactly the same way as real objects" (see, for example, column 9, lines 38-46). In other words, the stub object and the real object have the same behavior. Meyer further discloses that both the stub object and the real object are objects of the same class (see, for example, column 9, lines 54-61). Therefore, in Meyer, the SX_NEW and SX_DELETE instructions (see, for example, column 10, lines 14-35) generate and destruct, respectively, stub objects of the class and real objects of the class.

Moreover, Martin discloses that multiple copies of the same object are created to ensure high availability (see, for example, column 9, lines 35-38).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-3, 5-7 and 9-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,684,955 to Meyer et al. (art of record, "Meyer") in view of U.S. Patent No. 6,438,746 to Martin (art of record, "Martin").

With respect to claim 1 (currently amended), Meyer discloses a compiler embodied on a medium to compile a source program in an object-oriented programming language (see, for example, the abstract), said compiler comprising:

(a) detecting that a certain class-type variable is contained in an execution statement to be executed in parallel or a certain class-type variable is specified in a parallelization directive as a class to be parallelized (see, for example, column 5, lines 28-45, which shows detecting class-type variables in an object-oriented program that are to be distributed and executed over a plurality of processes);

(b) generating an instruction to call a construction instruction routine for an object of the class upon the detection, before said execution statement, in order to generate said object in addition to an original object of the class (see, for example, column 9, lines 38-46, which shows stub objects in addition to the original, real objects that both have the same behavior, and column 9, lines 54-61, which shows that both the stub objects and the original, real objects are objects of the same class, and see, for example, column 10, lines 14-35, which shows generating an SX_NEW instruction to call an instantiation function, i.e. a construction instruction routine, to generate the objects before execution); and

(c) generating an instruction to call a destruction instruction routine for the generated object of the class upon the detection, after said execution statement, in order to destruct the generated object in addition to said original object of the class (see, for example, column 9, lines 38-46, which shows stub objects in addition to the original, real objects that both have the same behavior, and column 9, lines 54-61, which shows that both the stub objects and the original, real objects are objects of the same class, and see, for example, column 10, lines 14-35, which shows

generating an SX_DELETE instruction to call a delete function, i.e. a destruction instruction routine, to destruct the objects after execution).

Although Meyer discloses that the object-oriented program is to be distributed and executed over a plurality of processes, as presented above, and further discloses that the object-oriented program may include calls or directives concerning the distribution of the class-type variables (see, for example, column 6, lines 1-10), Meyer does not expressly disclose the limitation wherein the execution statement is to be executed in parallel or is to be parallelized as specified in a parallelization directive.

However, Martin discloses an analogous compiler for a distributed object system (see, for example, the abstract), and discloses source code that includes class-type variables and comments (see, for example, column 7, lines 34-39). The comments are specifications for parallelization, and are thus parallelization directives for classes to be parallelized (see, for example, column 7, line 50 to column 8, line 5, and column 8, lines 54-63). When detecting that a certain class-type variable is to be parallelized and executed in parallel, the system generates constructor code to instantiate a number of copies of the same object in addition to the original object (see, for example, column 9, lines 44-54). This is done to ensure high availability of the object (see, for example, column 9, lines 35-38). Martin further discloses that the system may be integrated with a conventional compiler (see, for example, column 2, lines 25-29).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the compiler of Meyer with the distributed object features of Martin, so that the object-oriented program to be distributed and executed over a plurality of processes, as

taught by Meyer, can be compiled for parallel execution and thus compiled to ensure high availability, as taught by Martin.

With respect to claim 2 (currently amended), the rejection of claim 1 is incorporated, and Meyer in view of Martin further discloses:

when generating an intermediate language from said source program (see, for example, column 6, lines 34-48, which shows generating a protocol information file, i.e. an intermediate language, from the source code),

(a) allocating a construction and destruction instruction information region in the intermediate language of the class in addition to a region for said construction instruction routine and said destruction instruction routine, when a class-type variable which has possibility to be executed in parallel is specified (see, for example, column 10, lines 36-44, which shows allocating in the protocol information file a region for method information, including constructors and destructors); and

(b) storing into said construction and destruction instruction information region, information concerning said construction instruction routine and said destruction instruction routine of an object of the class (see, for example, column 10, lines 45-59, which shows storing information concerning the methods or routines of the classes), and

wherein information stored in said construction and destruction instruction information region is used in said steps of generating said instruction to call said construction instruction routine and generating of said instruction to call said destruction instruction routine (see, for example, column 10, lines 60-65, which shows that the method information is used when generating the instructions).

With respect to claim 3 (previously presented), the rejection of claim 2 is incorporated, and Meyer in view of Martin further discloses the limitation wherein said construction and destruction instruction information region is linked from a type information region storing a construction and destruction instruction information region index, and said type information region is linked from a class information region storing a type information region index (see, for example, column 10, lines 48-59, which shows class information linked to parameter type information linked to the method information, and which shows that the information is indexed by name and by a method identification), and when a class is identified, an access is performed from said class information region to said construction and destruction instruction information region via said type information region (see, for example, column 10, lines 36-48, which shows accessing the information to identify a class and its methods, including constructors and destructors).

With respect to claim 5 (currently amended), the limitations recited in claim 5 are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 6 (currently amended), the limitations recited in claim 6 are analogous to those of claim 2 (see the rejection of claim 2 above).

With respect to claim 7 (previously presented), the limitations recited in claim 7 are analogous to those of claim 3 (see the rejection of claim 3 above).

With respect to claim 9 (currently amended), the limitations recited in claim 9 are analogous to those of claim 1 (see the rejection of claim 1 above).

With respect to claim 10 (currently amended), the limitations recited in claim 10 are analogous to those of claim 2 (see the rejection of claim 2 above).

With respect to claim 11 (previously presented), the limitations recited in claim 11 are analogous to those of claim 3 (see the rejection of claim 3 above).

6. Claims 4, 8 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Meyer in view of Martin, as applied to claims 1, 5 and 9 above, respectively, and further in view of U.S. Patent No. 5,452,461 to Umekita et al. (art of record, "Umekita").

With respect to claim 4 (original), although Meyer discloses compiling an object-oriented program to distribute it over a plurality of processes (see, for example, column 5, lines 28-32 and 46-47), and although Martin discloses a compiler for a distributed system (see, for example, column 4, lines 15-20), Meyer in view of Martin does not expressly disclose the limitation wherein said compiler is a compiler for a parallel computer with shared memory.

However, Umekita discloses a compiler for a parallel computer with shared memory (see, for example, parallel compiler 213 in FIG. 27, and column 5, lines 30-27, which shows a computer having a plurality of processors and shared memory), for parallelizing a source program so that it may be executed in parallel with high efficiency (see, for example, column 1, lines 41-53).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement the system of Meyer and Martin in a parallel computer with shared memory, such as taught by Umekita, so as to achieve high efficiency.

With respect to claims 8 (currently amended) and 12 (original), the limitations recited in claims 8 and 12 are analogous to those of claim 4 (see the rejection of claim 4 above).

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER